**Frontend Assignment Set**

**Module 1 – Foundation**

**THEORY EXERCISE:**

**1.** What is a HTTP?

* **HTTP** stands for **HyperText Transfer Protocol**. It is the **foundation of data communication on the web.**

**2.** What is a Browsers? How they works?

* A **browser** is a **software application** that allows you to **access, retrieve, and view information on the World Wide Web.**

**3**. What is Domain Name?

* A **domain name** is a **human-readable name** (like google.com) that points to the **IP address** of a website.

**4.** What is hosting?

* **Hosting** means **putting your website on a server** so it can be seen online by anyone.

**Module 2 – Fundamentals of World Wide Web**

**THEORY EXERCISE:**

1. Difference between Web Designer and Web Developer

* **Web Designer** focuses on the **visual appearance and user experience** of a website, using tools like Figma and Photoshop.
* **Web Developer** focuses on the **functionality and coding** of the website, using languages like HTML, CSS, JavaScript, and more.

1. What is a W3C?

* **W3C (World Wide Web Consortium)** is the organization that creates web standards to ensure websites work correctly and consistently across all browsers and devices.

1. What is Domain?

* A **domain** is the **unique name** of a website that users type in the browser to access it (e.g., **google.com**). It represents the website’s address on the internet.

1. What SEO?

* **SEO (Search Engine Optimization)** is the process of improving a website to **rank higher in search engine results** and increase **organic (non-paid) traffic.**

1. What is SDLC life cycle?

* **SDLC (Software Development Life Cycle)** is a process used to **design, develop, test, and deploy software** in a structured way.

**Module 3 – Fundamentals of IT**

**What is a Program?**

1. Explain in your own words what a program is and how it functions.

* A program is a set of instructions that tells a computer what to do. It takes input, processes it using those instructions, and produces output, with the CPU executing the steps very quickly.

**What is Programming?**

1. What are the key steps involved in the programming process?

* **Define the problem** – Clearly understand what needs to be solved.
* **Plan the solution** – Break the problem into logical steps (often using flowcharts or pseudo code).
* **Write the code** – Implement the plan in a programming language.
* **Test the program** – Run it to find and fix errors (debugging).
* **Execute and review** – Ensure it meets the requirements and works efficiently.
* **Maintain the program** – Update or fix it as needs change.

**Types of Programming Languages**

1. What are the main differences between high-level and low-level programming languages?

* High-level languages are human-readable, portable, and easier to use (e.g., Python, Java), while low-level languages are hardware-specific, harder to read, but faster (e.g., Assembly, machine code).

**World Wide Web & How Internet Works**

1. Describe the roles of the client and server in web communication.

* **Client** – Requests information or services from a server (e.g., your web browser sending a page request).
* **Server** – Receives the request, processes it, and sends back the requested data or response (e.g., a website’s files or database results).

**Network Layers on Client and Server**

1. Explain the function of the TCP/IP model and its layers.

* The TCP/IP model defines how data is transmitted over a network, ensuring devices can communicate reliably.

**Its four layers and functions:**

1. Application layer – Provides services for user applications (e.g., HTTP for web, SMTP for email).
2. Transport layer – Ensures reliable data delivery using TCP or fast delivery using UDP.
3. Internet layer – Handles addressing and routing of data packets (e.g., IP).
4. Network Access layer – Manages physical data transmission over the network (e.g., Ethernet, Wi-Fi).

**Client and Servers**

1. Explain Client Server Communication

* Client-server communication is a fundamental networking concept where a client, like a web browser, sends a request to a server, which then processes the request and sends back a response.

**Types of Internet Connections**

1. How does broadband differ from fiber-optic internet?

* **Broadband** is a general term for high-speed internet delivered through various technologies like DSL, cable, or wireless.  
  **Fiber-optic internet** is a specific type of broadband that uses light signals through fiber-optic cables, offering much faster speeds, lower latency, and more reliability than traditional copper-based broadband.

**Protocols**

1. What are the differences between HTTP and HTTPS protocols?

* HTTP is an insecure protocol, transmitting data as plain text, making it vulnerable to interception and tampering. HTTPS, on the other hand, is a secure protocol that encrypts data using SSL/TLS, protecting sensitive information from eavesdropping and ensuring data integrity.

**Application Security**

1. What is the role of encryption in securing application, Software Applications and Its Types

* **Role of Encryption in Securing Applications:**  
  Encryption converts data into an unreadable format so that only authorized parties with the correct key can read it. In applications, it protects sensitive information (like passwords, payment details, or personal data) from hackers during storage or transmission.

**Software Applications and Their Types:**

1. **System Software** – Manages hardware and runs the computer (e.g., Operating Systems, Utilities).
2. **Application Software** – Performs specific user tasks (e.g., MS Word, Photoshop).
3. **Programming Software** – Helps developers create other software (e.g., compilers, IDEs).
4. **Web Applications** – Run on web browsers (e.g., Gmail, Google Docs).
5. **Mobile Applications** – Run on smartphones/tablets (e.g., WhatsApp, Instagram).
6. **Enterprise Applications** – For business operations (e.g., ERP, CRM systems).
7. What is the difference between system software and application software?

* **System Software** manages and controls computer hardware so other software can run (e.g., Operating Systems, device drivers).  
  **Application Software** is designed to help users perform specific tasks (e.g., MS Word, web browsers).

**Software Architecture**

1. What is the significance of modularity in software architecture?

* Modularity in software architecture is significant because it enables breaking down complex systems into smaller, manageable, and independent components (modules). This approach enhances various aspects of software development and maintenance, including improved organization, reusability, testing, scalability, and flexibility.

**Layers in Software Architecture**

1. Why are layers important in software architecture?

* Layers in software architecture are important because they **organize code into separate levels** with specific responsibilities, making the system easier to **understand, maintain, test, and scale**. They also allow **independent development** of parts and make it easier to update or replace one layer without affecting others.

**Software Environments**

1. Explain the importance of a development environment in software production.

* A development environment provides the tools and setup needed to build, test, and debug software efficiently, ensuring consistency, productivity, and safe testing before release.

**Source Code**

1. What is the difference between source code and machine code?

* **Source code** is the human-readable set of instructions written in a programming language (e.g., Python, Java).  
  **Machine code** is the binary (0s and 1s) form of those instructions that a computer’s CPU can directly execute.

**Github and Introductions**

1. Why is version control important in software development?

* Version control is important because it tracks changes to code, allows multiple developers to work together without overwriting each other’s work, makes it easy to revert to previous versions, and helps manage different features or bug fixes in parallel.

**Student Account in Github**

1. What are the benefits of using Github for students?

* GitHub helps students track code changes, collaborate on projects, build a portfolio, learn from open-source work, and access free tools through the Student Developer Pack.

**Types of Software**

1. What are the differences between open-source and proprietary software?

* Open-source software has publicly available code that anyone can use, modify, and share, often for free.  
  Proprietary software has closed code controlled by the owner, with limited rights for users and usually requires payment.

**GIT and GITHUB Training**

1. How does GIT improve collaboration in a software development team?

* Git improves collaboration by allowing multiple developers to work on the same project simultaneously without overwriting each other’s work.  
  It tracks changes, manages versions, enables branching and merging, and makes it easy to review, share, and integrate code efficiently.

**Application Software**

1. What is the role of application software in businesses?

* Application software in businesses helps perform specific tasks such as managing data, processing transactions, creating documents, analyzing information, and communicating.  
  It improves efficiency, accuracy, and productivity, enabling better decision-making and smoother operations.

**Software Development Process**

1. What are the main stages of the software development process?

* The software development process, also known as the Software Development Life Cycle (SDLC), typically includes these main stages: Planning, Analysis, Design, Development, Testing, Deployment, and Maintenance.

**Software Requirement**

1. Why is the requirement analysis phase critical in software development?

* The requirement analysis phase is critical because it defines exactly what the software should do, ensuring that developers build the right product.  
  It helps prevent misunderstandings, reduces costly changes later, and aligns the final product with the client’s needs and goals.

**Software Analysis**

1. What is the role of software analysis in the development process?

* Software analysis identifies and defines the requirements, constraints, and goals of a project.  
  It ensures a clear understanding of what needs to be built, guides design and development decisions, and helps avoid errors, rework, and scope creep later in the process.

**System Design**

1. What are the key elements of system design?

* Architecture Design, **Data Design,** Interface Design, Component Design, Security Design, Performance Design.

**Software Testing**

1. Why is software testing important?

* Software testing is important because it ensures the software works correctly, meets requirements, and is free from critical bugs.  
  It improves quality, increases reliability, enhances user satisfaction, and helps prevent costly failures after release.

**Maintenance**

1. What types of software maintenance are there?

* Corrective Maintenance, Adaptive Maintenance, Perfective Maintenance, **Preventive Maintenance.**

**Development**

1. What are the key differences between web and desktop applications?

* Web applications run in a browser, need no installation, and are accessible anywhere with internet, while desktop applications require installation, run locally, and often work without internet.

**Web Application**

1. What are the advantages of using web applications over desktop applications?

* Web applications are accessible from anywhere, need no installation, update automatically, work on any platform, and support easy real-time collaboration.

**Designing**

1. What role does UI/UX design play in application development?

* UI/UX design ensures an application is visually appealing, easy to use, and provides a smooth, satisfying user experience.  
  It improves usability, boosts user satisfaction, and increases engagement, leading to better adoption and success of the application.

**Mobile Application**

1. What are the differences between native and hybrid mobile apps?

* In a native app, your developers have to rewrite and redesign all the app functionality in the native development language. A hybrid app lets you write the app functionality in a single codebase. You can then wrap your code in a lightweight native app shell or container.

**DFD (Data Flow Diagram)**

1. What is the significance of DFDs in system analysis?

* Data Flow Diagrams (DFDs) are significant in system analysis because they visually represent how data moves through a system.  
  They help analysts understand processes, data inputs/outputs, and storage points, making it easier to identify requirements, inefficiencies, and improvements.

**Desktop Application**

1. What are the pros and cons of desktop applications compared to web applications?

* **Pros:** Faster performance, works offline, full hardware access, richer features.  
  **Cons:** Requires installation, platform-specific, manual updates, limited to installed devices.

**Flow Chart**

1. How do flowcharts help in programming and system design?

* Flowcharts help by visually mapping the steps, decisions, and processes in a system or program.  
  They make logic easier to understand, identify errors or inefficiencies, and serve as a clear guide for development and communication among team members.

**Module 2 – Frontend – HTML**

**HTML Basics**

**Theory Assignment**

1. Define HTML. What is the purpose of HTML in web development?

* HTML (Hypertext Markup Language) is the standard language used to create and structure content on the web.  
  Its purpose in web development is to define the layout and elements of a webpage—such as text, images, links, and forms—so browsers can display them properly.

1. Explain the basic structure of an HTML document. Identify the mandatory tags and their purposes.

* <!DOCTYPE html>

<html>

<head>

<title>Page Title</title>

</head>

<body>

<h1>Hello, World!</h1>

<p>This is a paragraph.</p>

</body></html>

**<!DOCTYPE html>** – Declares the document type and HTML version (HTML5).

**<html>** – Root element that contains all HTML content.

**<head>** – Contains metadata about the page (title, styles, scripts, etc.).

**<title>** – Sets the title displayed in the browser tab.

**<body>** – Contains the visible content of the webpage.

1. What is the difference between block-level elements and inline elements in HTML? Provide examples of each.

* **Block-level Elements:**

Always start on a new line.

By default, they take up the full available width of their parent container, even if their content doesn't fill that space.

You can set their width, height, margin (all sides), and padding (all sides).

Can contain both other block-level elements and inline elements.

**Examples:**

<div>: A generic container for grouping other elements.

<p>: A paragraph of text.

<h1> to <h6>: Headings of different levels.

<ul>, <ol>, <li>: Lists and list items.

<header>, <footer>, <nav>, <section>, <article>: Semantic elements for page structure.

* **Inline Elements:**

Do not start on a new line; they flow horizontally within the current line of content.

Only take as much width as necessary for their content.

You cannot set their width or height directly (they are determined by their content).

Only respect horizontal margin and padding (left and right); vertical margin and padding often have no visual effect or behave unpredictably.

Can generally only contain data or other inline elements. They should not contain block-level elements.

**Examples:**

<a>: A hyperlink (link).

<span>: A generic inline container for styling small sections of text.

<strong> / <b>: For bold text.

<img>: An image.

<em> / <i>: For italicized/emphasized text.

1. Discuss the role of semantic HTML. Why is it important for accessibility and SEO? Provide examples of semantic elements.

* **Role of Semantic HTML:**  
  Semantic HTML uses tags that clearly describe the meaning and purpose of the content they contain, making web pages more structured and meaningful.

**Importance:**

* **Accessibility:** Helps screen readers and assistive technologies understand the page structure, improving the browsing experience for users with disabilities.
* **SEO:** Search engines better understand content hierarchy and relevance, which can improve search rankings.
* **Maintainability:** Makes code more readable and easier to maintain.

**Examples of Semantic Elements:**

<header> – Defines the top section of a page or section.

<nav> – Contains navigation links.

<main> – Specifies the main content area.

<article> – Represents an independent piece of content.

<section> – Groups related content together.

<footer> – Defines the bottom section of a page or section.

**HTML Forms**

**Theory Assignment**

1. What are HTML forms used for? Describe the purpose of the input, textarea, select, and button elements.

* **HTML Forms** are used to collect user input and send it to a server for processing, such as login details, search queries, or feedback.

<input>: Takes single-line input like text, numbers, emails.

<textarea>: Takes multi-line text input.

<select>: Creates a drop-down list of options.

<button>: Submits the form or performs an action.

1. Explain the difference between the GET and POST methods in form submission. When should each be used?

* **GET** sends form data appended in the URL, making it visible and limited in size; it’s best for retrieving data like search queries.

**POST** sends data inside the request body, keeping it hidden and allowing larger amounts; it’s used for submitting sensitive info or when changing server data, like login or form submissions.

1. What is the purpose of the label element in a form, and how does it improve accessibility?

* The <label> element associates a text description with a form control (like an input).

**It improves accessibility by:**

* Helping screen readers announce the purpose of the form field clearly.
* Enlarging the clickable area when linked properly, making forms easier to use.

You link a label to an input using the for attribute matching the input’s id.

**HTML Tables**

**Theory Assignment**

1. Explain the structure of an HTML table and the purpose of each of the following elements: <table>, <tr>, <th>, <td> and <thead>.

* <table> : This is the container element for the entire table.

<tr> : Represents a single row in the table.

<th> : Defines a header cell for a column or row.

<td> : Represents a standard data cell in the table.

<thead> : Groups the header content of a table.

1. What is the difference between colspan and rowspan in tables? Provide examples.

* Colspan: Column span — it makes a single cell stretch **across multiple columns.**

**Example:** <td colspan="2"> → spans 2 columns.

* Rowspan: Row span — it makes a single cell stretch **across multiple rows.**

**Example:** <td rowspan="3"> → spans 3 rows.

1. Why should tables be used sparingly for layout purposes? What is a better alternative?

* Tables should be used sparingly for layout because they harm accessibility, break semantic meaning, are hard to maintain, and don’t adapt well to responsive design.  
  **Better alternative:** Use **CSS Flexbox** or **CSS Grid** for layout.